FIG. 1

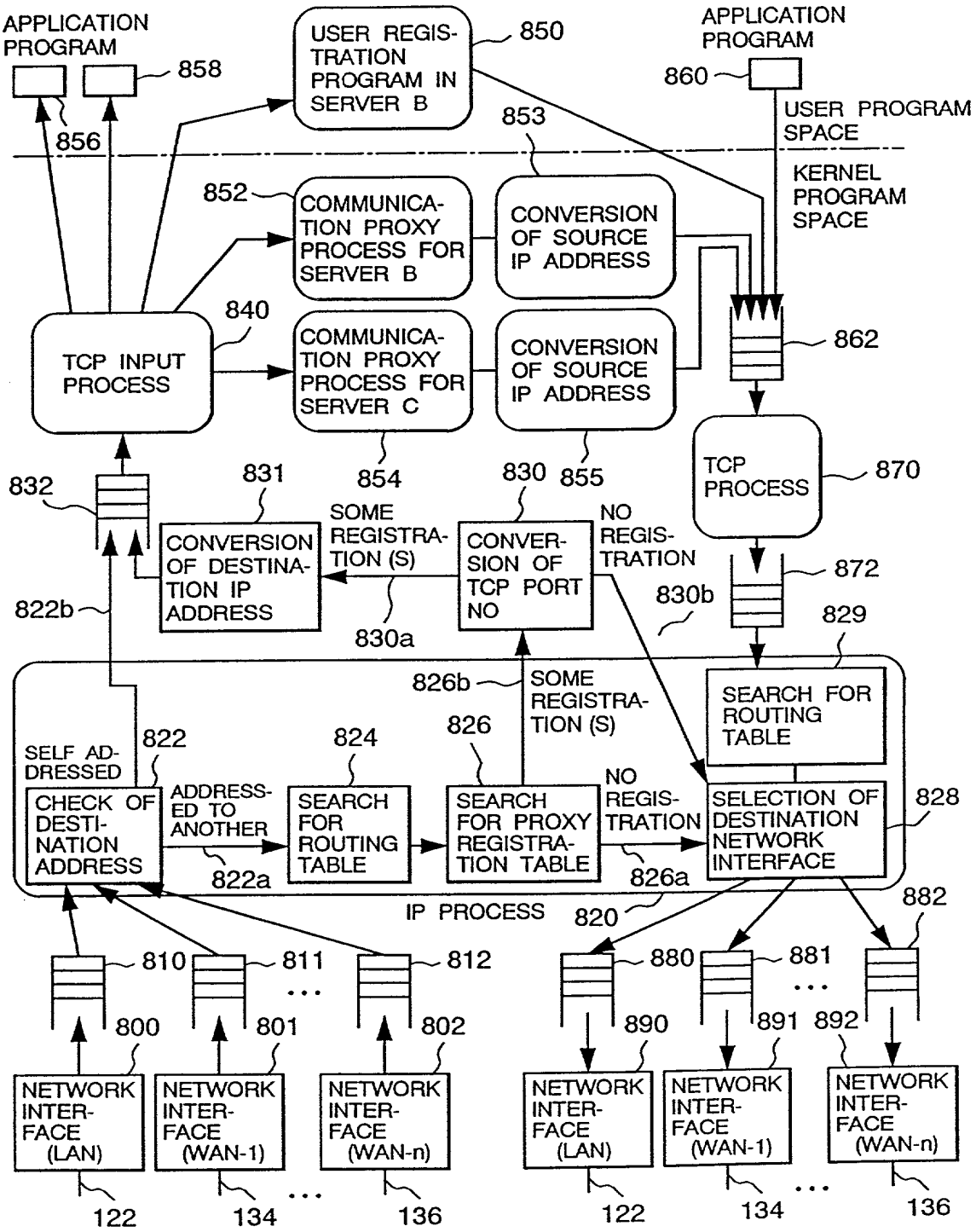

APPLICATION PROGRAM
858
856
USER REGISTRATION PROGRAM IN SERVER B
850
APPLICATION PROGRAM
860
USER PROGRAM SPACE
853
KERNEL PROGRAM SPACE
852
COMMUNICATION PROXY PROCESS FOR SERVER B
CONVERSION OF SOURCE IP ADDRESS
840
TCP INPUT PROCESS
COMMUNICATION PROXY PROCESS FOR SERVER C
CONVERSION OF SOURCE IP ADDRESS
862
TCP PROCESS
870
832
831
854
830
855
CONVERSION OF DESTINATION IP ADDRESS
SOME REGISTRATION (S)
CONVERSION OF TCP PORT NO
NO REGISTRATION
872
822b
830a
830b
829
826b
SOME REGISTRATION (S)
SEARCH FOR ROUTING TABLE
SELF ADDRESSED
822
824
826
CHECK OF DESTINATION ADDRESS
ADDRESSED TO ANOTHER
SEARCH FOR ROUTING TABLE
SEARCH FOR PROXY REGISTRATION TABLE
NO REGISTRATION
SELECTION OF DESTINATION NETWORK INTERFACE
828
822a
826a
IP PROCESS
820
882
810
811
812
880
881
800
801
802
890
891
892
NETWORK INTERFACE (LAN)
NETWORK INTERFACE (WAN-1)
NETWORK INTERFACE (WAN-n)
NETWORK INTERFACE (LAN)
NETWORK INTERFACE (WAN-1)
NETWORK INTERFACE (WAN-n)
122
134
136
122
134
136

FIG. 2

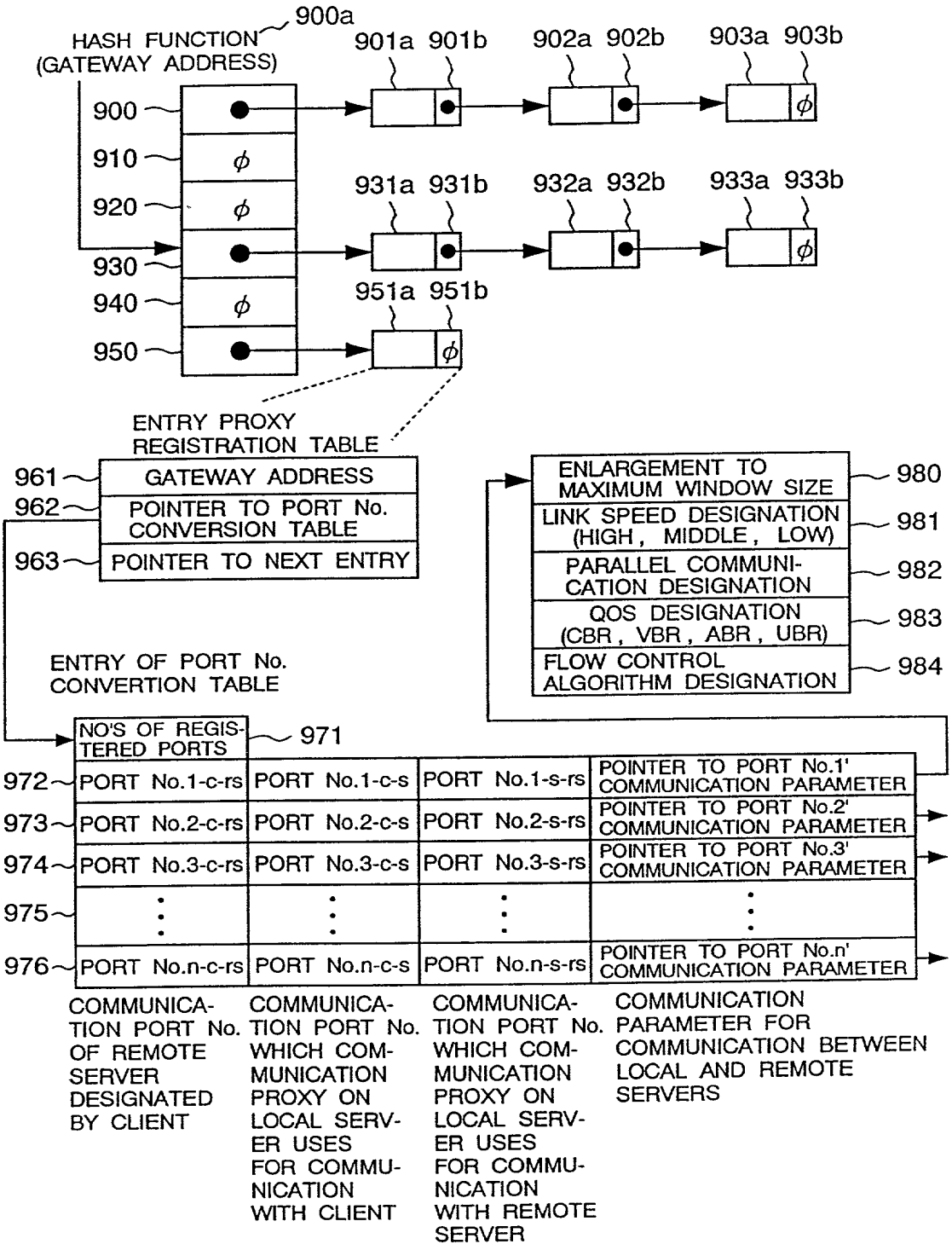

900a
HASH FUNCTION (GATEWAY ADDRESS)
900
910
920
930
940
950
901a 901b
902a 902b
903a 903b
931a 931b
932a 932b
933a 933b
951a 951b
ENTRY PROXY REGISTRATION TABLE
961 GATEWAY ADDRESS
962 POINTER TO PORT No. CONVERSION TABLE
963 POINTER TO NEXT ENTRY
ENLARGEMENT TO MAXIMUM WINDOW SIZE 980
LINK SPEED DESIGNATION (HIGH, MIDDLE, LOW) 981
PARALLEL COMMUNICATION DESIGNATION 982
QOS DESIGNATION (CBR, VBR, ABR, UBR) 983
FLOW CONTROL ALGORITHM DESIGNATION 984
ENTRY OF PORT No. CONVERTION TABLE
NO'S OF REGISTERED PORTS 971
972 PORT No.1-c-rs | PORT No.1-c-s | PORT No.1-s-rs | POINTER TO PORT No.1' COMMUNICATION PARAMETER
973 PORT No.2-c-rs | PORT No.2-c-s | PORT No.2-s-rs | POINTER TO PORT No.2' COMMUNICATION PARAMETER
974 PORT No.3-c-rs | PORT No.3-c-s | PORT No.3-s-rs | POINTER TO PORT No.3' COMMUNICATION PARAMETER
975
976 PORT No.n-c-rs | PORT No.n-c-s | PORT No.n-s-rs | POINTER TO PORT No.n' COMMUNICATION PARAMETER
COMMUNICATION PORT No. OF REMOTE SERVER DESIGNATED BY CLIENT
COMMUNICATION PORT No. WHICH COMMUNICATION PROXY ON LOCAL SERVER USES FOR COMMUNICATION WITH CLIENT
COMMUNICATION PORT No. WHICH COMMUNICATION PROXY ON LOCAL SERVER USES FOR COMMUNICATION WITH REMOTE SERVER
COMMUNICATION PARAMETER FOR COMMUNICATION BETWEEN LOCAL AND REMOTE SERVERS

FIG. 3

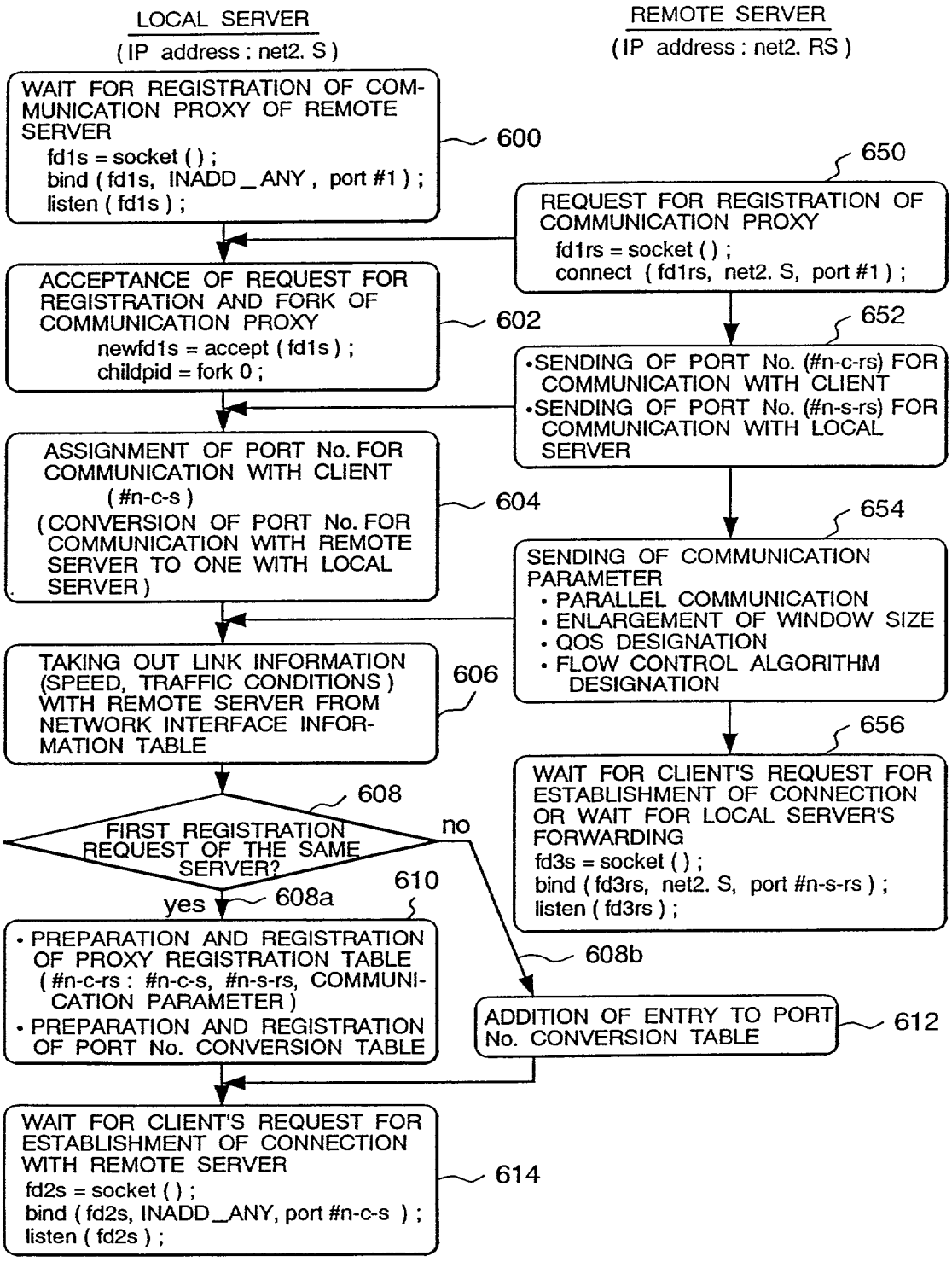

LOCAL SERVER
(IP address : net2. S)
REMOTE SERVER
(IP address : net2. RS)
WAIT FOR REGISTRATION OF COMMUNICATION PROXY OF REMOTE SERVER
fd1s = socket ( ) ;
bind (fd1s, INADD_ANY , port #1) ;
listen (fd1s) ;
600
650
REQUEST FOR REGISTRATION OF COMMUNICATION PROXY
fd1rs = socket ( ) ;
connect (fd1rs, net2. S, port #1) ;
ACCEPTANCE OF REQUEST FOR REGISTRATION AND FORK OF COMMUNICATION PROXY
newfd1s = accept (fd1s) ;
childpid = fork 0 ;
602
652
•SENDING OF PORT No. (#n-c-rs) FOR COMMUNICATION WITH CLIENT
•SENDING OF PORT No. (#n-s-rs) FOR COMMUNICATION WITH LOCAL SERVER
ASSIGNMENT OF PORT No. FOR COMMUNICATION WITH CLIENT (#n-c-s)
(CONVERSION OF PORT No. FOR COMMUNICATION WITH REMOTE SERVER TO ONE WITH LOCAL SERVER)
604
654
SENDING OF COMMUNICATION PARAMETER
• PARALLEL COMMUNICATION
• ENLARGEMENT OF WINDOW SIZE
• QOS DESIGNATION
• FLOW CONTROL ALGORITHM DESIGNATION
TAKING OUT LINK INFORMATION (SPEED, TRAFFIC CONDITIONS) WITH REMOTE SERVER FROM NETWORK INTERFACE INFORMATION TABLE
606
656
WAIT FOR CLIENT'S REQUEST FOR ESTABLISHMENT OF CONNECTION OR WAIT FOR LOCAL SERVER'S FORWARDING
fd3s = socket ( ) ;
bind (fd3rs, net2. S, port #n-s-rs) ;
listen (fd3rs) ;
608
FIRST REGISTRATION REQUEST OF THE SAME SERVER?
no
yes
608a
610
• PREPARATION AND REGISTRATION OF PROXY REGISTRATION TABLE (#n-c-rs : #n-c-s, #n-s-rs, COMMUNICATION PARAMETER)
• PREPARATION AND REGISTRATION OF PORT No. CONVERSION TABLE
608b
ADDITION OF ENTRY TO PORT No. CONVERSION TABLE
612
WAIT FOR CLIENT'S REQUEST FOR ESTABLISHMENT OF CONNECTION WITH REMOTE SERVER
fd2s = socket ( ) ;
bind (fd2s, INADD_ANY, port #n-c-s) ;
listen (fd2s) ;
614

FIG. 4

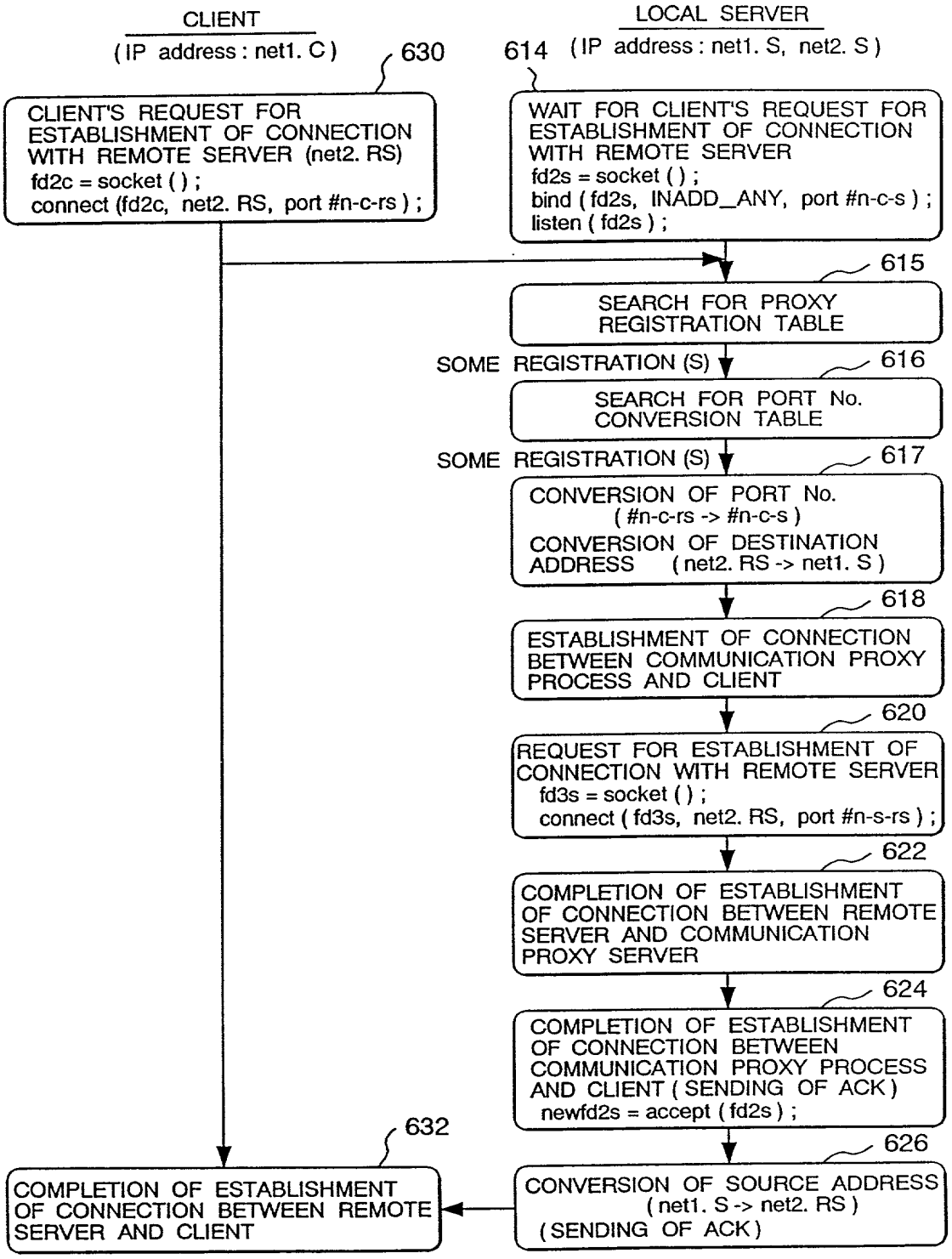

CLIENT
(IP address : net1. C)
630
CLIENT'S REQUEST FOR ESTABLISHMENT OF CONNECTION WITH REMOTE SERVER (net2. RS)
fd2c = socket ( ) ;
connect (fd2c, net2. RS, port #n-c-rs) ;
LOCAL SERVER
614
(IP address : net1. S, net2. S)
WAIT FOR CLIENT'S REQUEST FOR ESTABLISHMENT OF CONNECTION WITH REMOTE SERVER
fd2s = socket ( ) ;
bind (fd2s, INADD_ANY, port #n-c-s) ;
listen (fd2s) ;
615
SEARCH FOR PROXY REGISTRATION TABLE
SOME REGISTRATION (S)
616
SEARCH FOR PORT No. CONVERSION TABLE
SOME REGISTRATION (S)
617
CONVERSION OF PORT No. (#n-c-rs -> #n-c-s)
CONVERSION OF DESTINATION ADDRESS (net2. RS -> net1. S)
618
ESTABLISHMENT OF CONNECTION BETWEEN COMMUNICATION PROXY PROCESS AND CLIENT
620
REQUEST FOR ESTABLISHMENT OF CONNECTION WITH REMOTE SERVER
fd3s = socket ( ) ;
connect (fd3s, net2. RS, port #n-s-rs) ;
622
COMPLETION OF ESTABLISHMENT OF CONNECTION BETWEEN REMOTE SERVER AND COMMUNICATION PROXY SERVER
624
COMPLETION OF ESTABLISHMENT OF CONNECTION BETWEEN COMMUNICATION PROXY PROCESS AND CLIENT (SENDING OF ACK)
newfd2s = accept (fd2s) ;
626
CONVERSION OF SOURCE ADDRESS (net1. S -> net2. RS)
(SENDING OF ACK)
632
COMPLETION OF ESTABLISHMENT OF CONNECTION BETWEEN REMOTE SERVER AND CLIENT

FIG. 5a

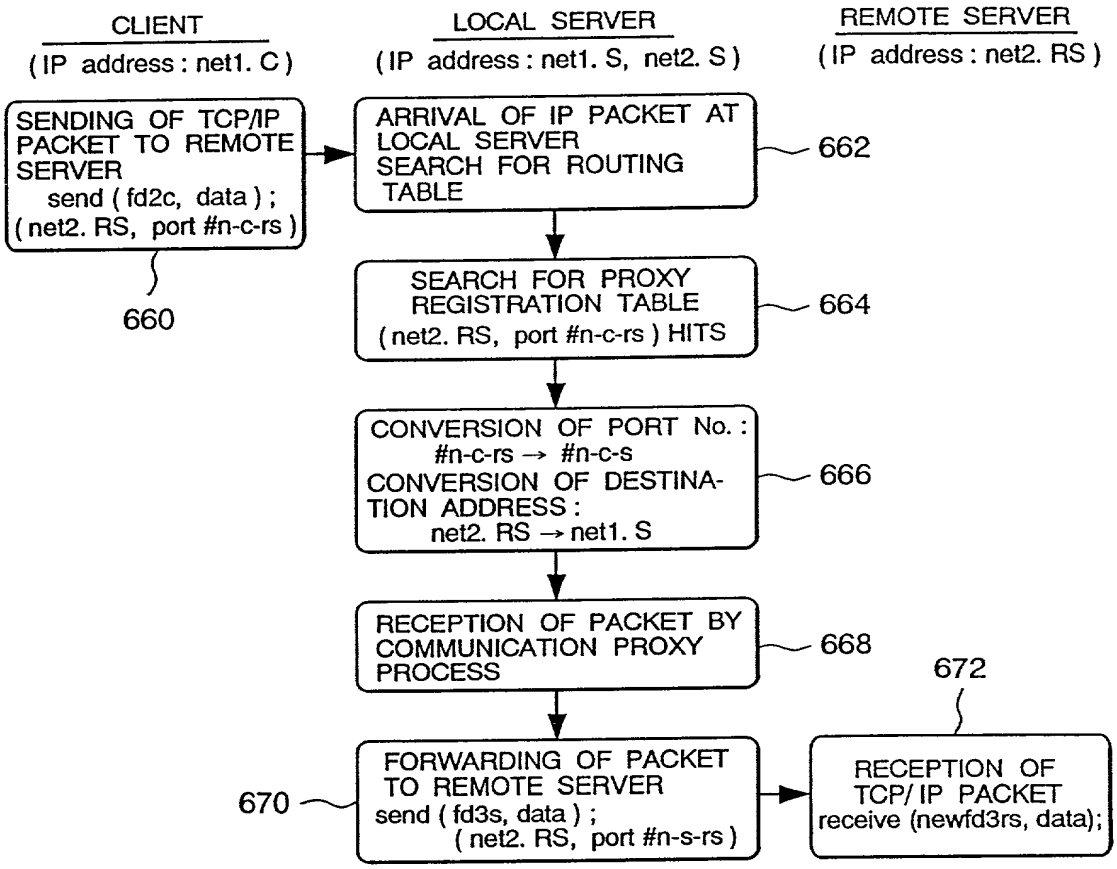
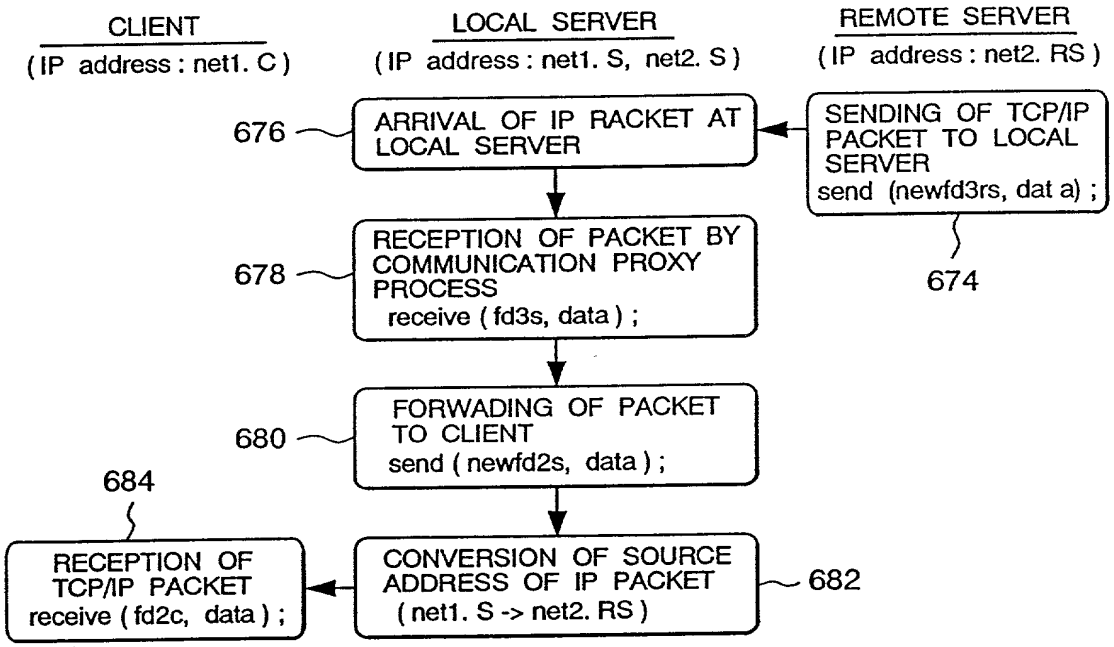

CLIENT
(IP address : net1. C)
LOCAL SERVER
(IP address : net1. S, net2. S)
REMOTE SERVER
(IP address : net2. RS)
SENDING OF TCP/IP PACKET TO REMOTE SERVER
send (fd2c, data) ;
(net2. RS, port #n-c-rs)
660
ARRIVAL OF IP PACKET AT LOCAL SERVER
SEARCH FOR ROUTING TABLE
662
SEARCH FOR PROXY REGISTRATION TABLE
(net2. RS, port #n-c-rs) HITS
664
CONVERSION OF PORT No. :
#n-c-rs → #n-c-s
CONVERSION OF DESTINATION ADDRESS :
net2. RS → net1. S
666
RECEPTION OF PACKET BY COMMUNICATION PROXY PROCESS
668
672
FORWARDING OF PACKET TO REMOTE SERVER
send (fd3s, data) ;
(net2. RS, port #n-s-rs)
670
RECEPTION OF TCP/IP PACKET
receive (newfd3rs, data);

FIG. 5b



EP 0 854 623 A2

PRIOR ART

FIG. 6a

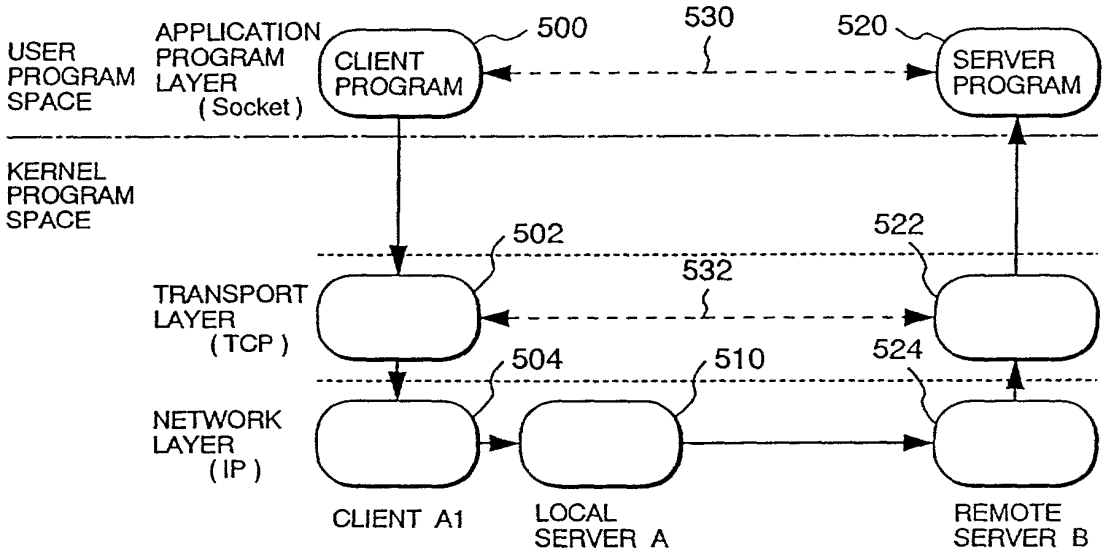


FIG. 6b

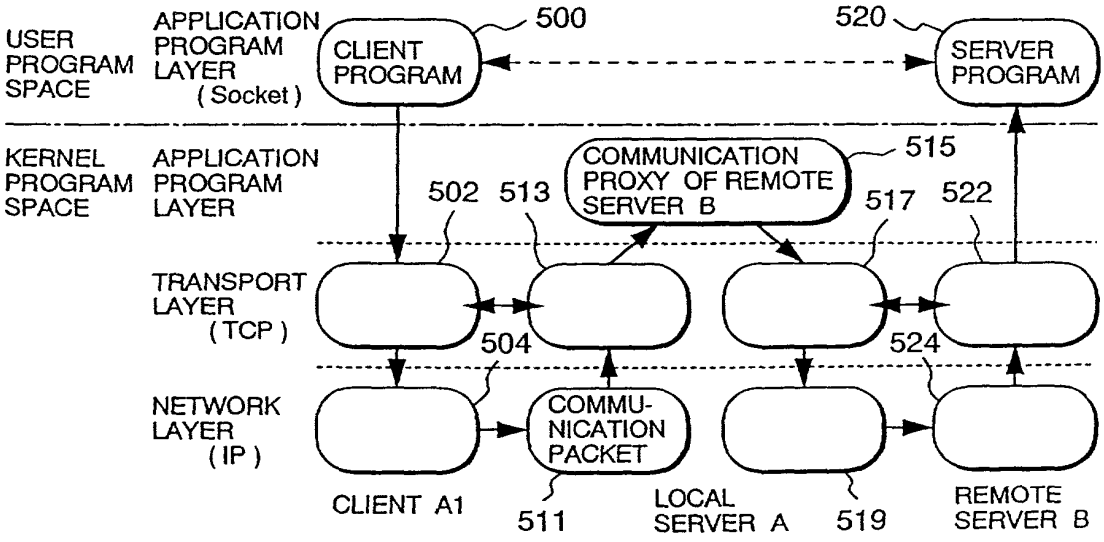


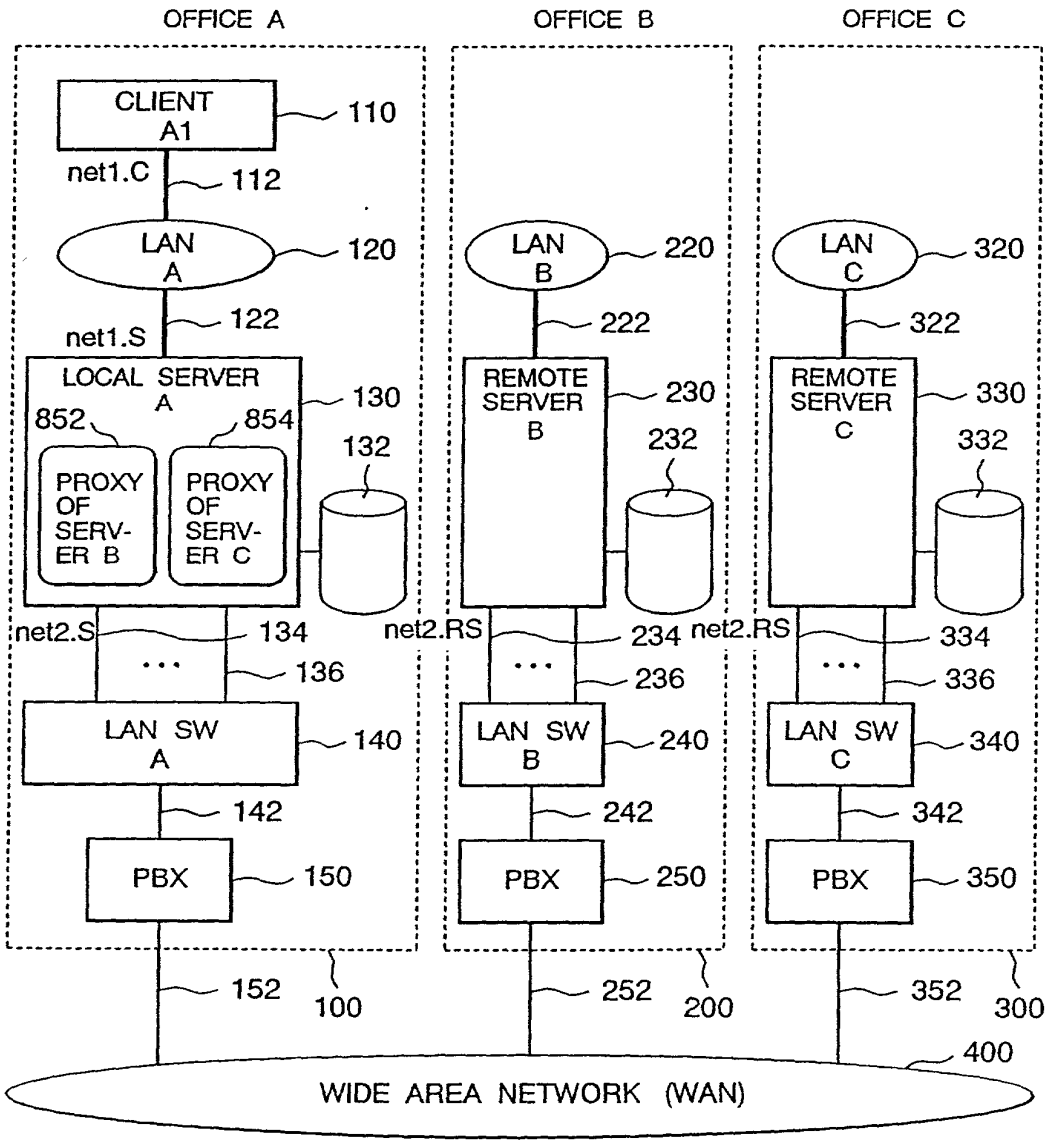
FIG. 7a

```
#define SERV_TCP_PORT    6001                              ~~ 701

fd = socket (AF_INET, SOCK_STREAM, 0) ;                    ~~ 702

serv_addr . sin_family = AF_INET ;                         ~~ 703
serv_addr . sin_addr. s_addr = htonl (INADDR ANY) ;        ~~ 704
serv_addr . sin_port = htons (SERV_TCP_PORT) ;             ~~ 705

bind (fd , (struct sockaddr *) & serv_addr ,               ~~ 706
      sizeof (serv_addr) ) ;

listen (fd , 5) ;                                          ~~ 707

for ( ; ; ) {
      newfd = accept (fd , (struct aockaddr *)             ~~ 708
                      &cli_addr , &clilen) ;               ~~ 709
      childpid = fork ( ) ;                                ~~ 710
      if (childpid == 0) {                                 ~~ 711
         close (fd) ;          /* child process */         ~~ 712
         send and receive data to and from CLIENT ;        ~~ 713
         exit (0) ;                                        ~~ 714
      }
      close (newfd) ;          /* parent process */        ~~ 715
```

FIG. 7b

```
#define SERV_TCP_PORT    6001                              ~~ 750
#define SERV_HOST_ADDR   net1. I                           ~~ 751

fd = socket (AF_INET , SOCK_STREAM , 0) ;                  ~~ 753

serv_addr . sin_family = AF_INET ;                         ~~ 754
serv_addr . sin_addr. s_addr = htonl (SERV_ HOST_ADDR) ;   ~~ 755
serv_addr . sin_port = htons (SERV_TCP_PORT) ;             ~~ 756

connect (fd , (struct sockaddr *) &serv_addr ,             ~~ 758
         sizeof (serv_addr) ) ;

send and receive data to and from SERVER ;                 ~~ 759

close (fd) ;                                               ~~ 760
exit (0) ;                                                 ~~ 761
```

FIG. 8



LAN : Local Area Network
WAN : Wide Area Network
SW : Switch
PBX : Private Branch Exchange

EP 0 952 524 A2